

## Integrating multiple predictions

### ModelRX Case Study

Tomaž Mohorič, Edelweiss Connect

OpenRiskNet: Open e-Infrastructure to Support Data Sharing, Knowledge Integration and *in silico* Analysis and Modelling in Risk Assessment  
Project Number 731075



# Problem

Given multiple evidences make a **consensus** prediction about a blood-brain barrier penetration and estimate its **uncertainty**

# Approach

**Dempster-Shafer theory** (DST) provides framework for such integrated risk assessment

# The idea behind DST - single evidence

*Will it rain tomorrow?*

- A friend predicts 70 % chance of rain (30 % chance of no-rain)
- We estimate his reliability to 60 %

*What shall we expect?*

- Belief of **rain**:  $0.7 \times 0.6 = \mathbf{0.42}$
- Belief of **no-rain**:  $0.3 \times 0.6 = \mathbf{0.18}$
- **Uncertainty**:  $1 - 0.6 = \mathbf{0.4}$

# The idea behind DST - single evidence

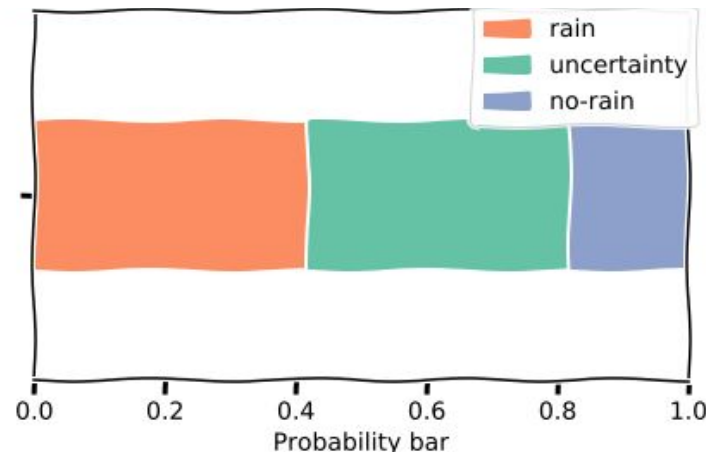
*Will it rain tomorrow?*

- A friend predicts 70 % chance of rain (30 % chance of no-rain)
- We estimate his reliability to 60 %

*What shall we expect?*

- Probability of **rain**: **0.42 - 0.82**
- Probability of **no-rain**: **0.18 - 0.58**
- **Uncertainty**: **0.4**

*DST prediction: **equivocal***



# The idea behind DST - multiple evidences

*Will it rain tomorrow?*

- A friend predicts 70 % chance of rain (60 % reliability)
- Newspaper predicts 60 % chance of rain (80 % reliability)
- Individual predictions are equivocal

*What shall we expect?*

- DST provides framework to combine such evidences into a consensus prediction

# The idea behind DST - multiple evidences

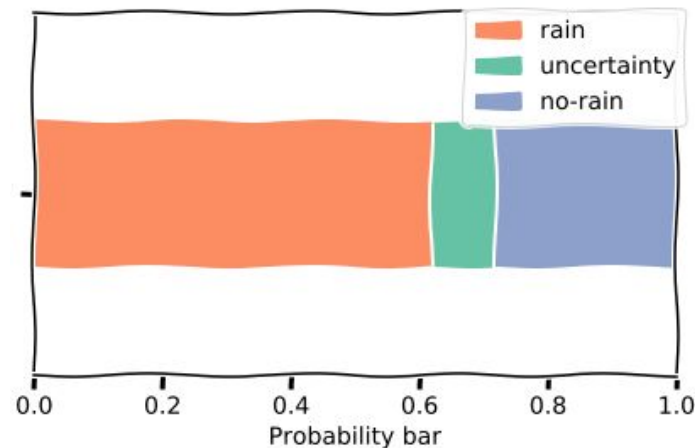
*Will it rain tomorrow?*

- A friend predicts 70 % chance of rain (60 % reliability)
- Newspaper predicts 60 % chance of rain (80 % reliability)
- Individual predictions are equivocal

*What shall we expect?*

- Probability of **rain**: **0.62 - 0.72**
- Probability of **no-rain**: **0.28 - 0.38**
- **Uncertainty**: **0.10**

*DST prediction: rain*

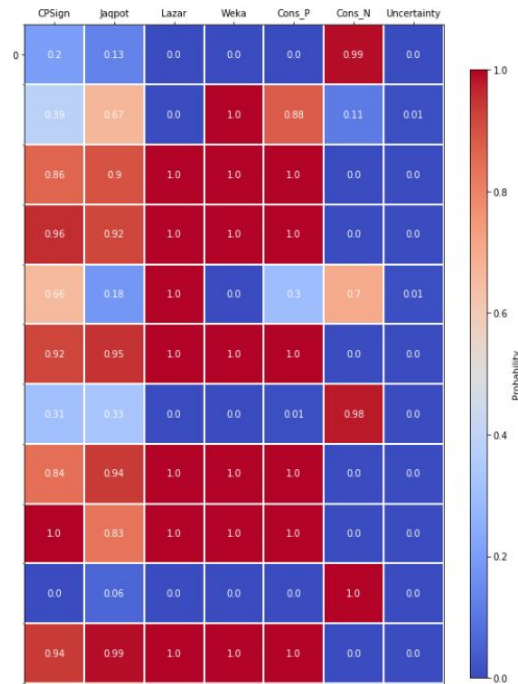


# ModelRX case study

- **Predictive models** for blood-brain barrier penetration:  
Lazar, Jaqpot, WEKA JGU, Venn-ABERS
- **Reliability** of each model (cross-validation):  
Positive predictive value (% true positive of all predicted positive)  
Negative predictive value (% true negative of all predicted negative)
- **Test set** of ~ 400 compounds

# ModelRX case study

- GitHub: [OpenRiskNet/workshop/ModelRX/Blood-brain barrier - Consensus](https://github.com/OpenRiskNet/workshop/ModelRX/Blood-brain%20barrier%20-%20Consensus)
- Jupyter notebook: [consensus-batch-offline.ipynb](#)
- Consensus predictions for 11 compounds of test set





# Hands-on exercise

- Consensus prediction using 2 predictive models (Lazar, CPSign)
- GitHub: [OpenRiskNet/workshop/ModelRX/Blood-brain barrier - Consensus](https://github.com/OpenRiskNet/workshop/ModelRX/Blood-brain%20barrier%20-%20Consensus)
- Jupyter notebook: [consensus-single-web-CL.ipynb](#)

## Main steps:

1. Provide compound structure (SMILES)
2. Access web services through REST API (Lazar, CPSign)
3. Make a consensus prediction using Python library [dst.py](#)

# Hands-on exercise

## 1. Provide compound structure (SMILES)

```
smiles = 'COCCC'
```

## 2. Access the Lazar API

```
r = requests.post(  
    url = 'https://lazar.prod.openrisknet.org/model/5ae2dd885f1c2d01323270ee',  
    data = {'identifier': smiles},  
    headers = {'accept': 'text/csv'}  
)  
  
result = json.loads(r.text)  
  
print('Prediction:', result['prediction']['value'])
```

```
Prediction: penetrating
```

# Hands-on exercise

## 3. Access the CPSign API

```
r = requests.get(
    url = "http://blood-brain-barrier-penetration-cvap-cpsign.prod.openrisknet.org/v1/predict",
    params = {'molecule': smiles},
    headers = {'accept': 'application/json'}
)

result = json.loads(r.text)

result['prediction']

[{'probability': 0.006, 'label': 'non-penetrating'},
 {'probability': 0.994, 'label': 'penetrating'}]
```

# Hands-on exercise

## 4. Make a consensus prediction

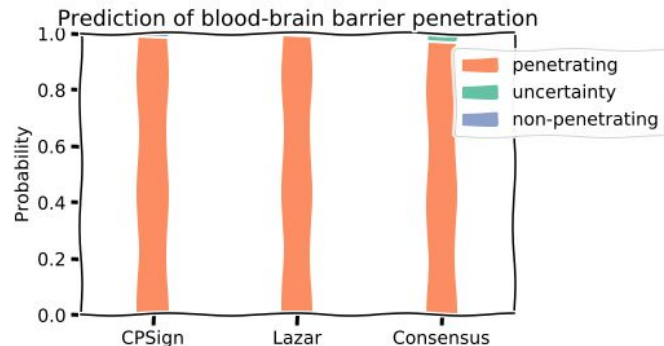
```
print('Prediction:\t', models_pred)
print('Probability:\t', models_prob)
print('PPV:\t\t', models_ppv)
print('NPV:\t\t', models_npv)
```

```
Prediction:      {'Lazar': 'P', 'CPSign': 'P'}
Probability:     {'Lazar': 1.0, 'CPSign': 0.994}
PPV:             {'CPSign': 0.809, 'Lazar': 0.886}
NPV:             {'CPSign': 0.701, 'Lazar': 0.489}
```

```
beliefs, plausibilities, result = dst.predict_Dempster(models_pred, models_prob, models_ppv, models_npv)
```

```
print('Beliefs:\t', beliefs)
print('Plausibilities:\t', plausibilities)
print('Result:\t\t', result)
```

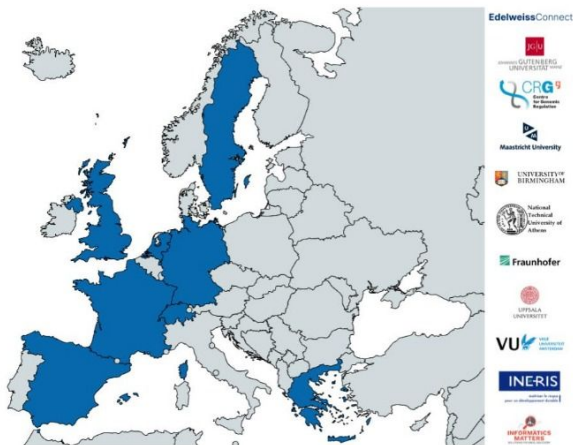
```
Beliefs:         {'P': 0.9775762072502536, 'N': 0.0005557460659842, 'PN': 1.0}
Plausibilities:  {'P': 0.9994442539340158, 'N': 0.022423792749746436, 'PN': 1.0}
Result:         P
```



# Acknowledgements

**OpenRiskNet** (Grant Agreement 731075) is a project funded by the European Commission within Horizon 2020 Programme

## Project partners:



- P1 Edelweiss Connect GmbH, Switzerland (EwC)
- P2 Johannes Gutenberg-Universität Mainz, Germany (JGU)
- P3 Fundacio Centre De Regulacio Genomica, Spain (CRG)
- P4 Universiteit Maastricht, Netherlands (UM)
- P5 The University Of Birmingham, United Kingdom (UoB)
- P6 National Technical University Of Athens, Greece (NTUA)
- P7 Fraunhofer Gesellschaft Zur Foerderung Der Angewandten Forschung E.V., Germany (Fraunhofer)
- P8 Uppsala Universitet, Sweden (UU)
- P9 Medizinische Universität Innsbruck, Austria (MUI)
- P10 Informatics Matters Limited, United Kingdom (IM)
- P11 Institut National De L'environnement Et Des Risques INERIS, France (INERIS)
- P12 Vrije Universiteit Amsterdam, Netherlands (VU)